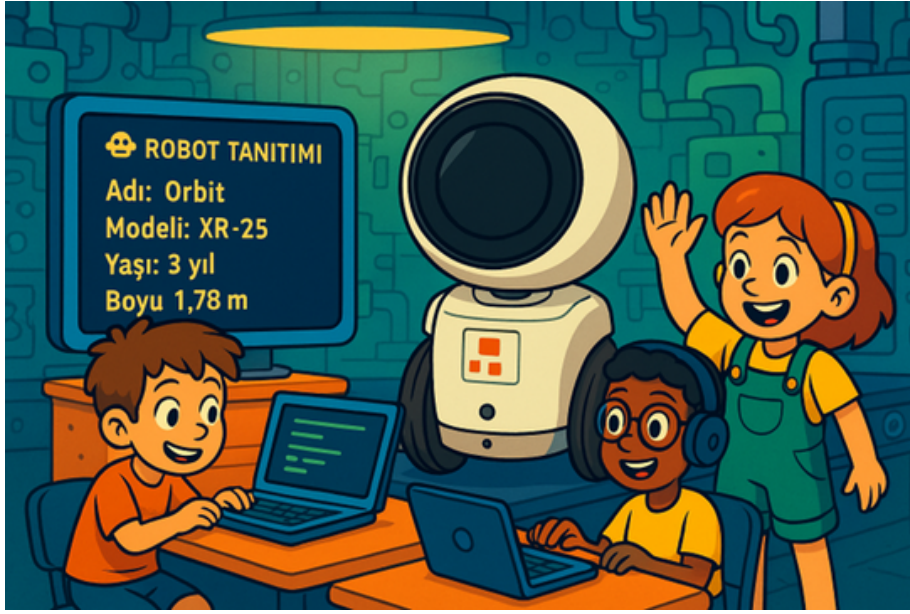


Değişkenler ve Veri Türleri

Robot Fabrikasında Macera: Orbit ile Tanışıyoruz

Sahne 1

Orbit'in Kimlik Kartını
Hazırlıyoruz

Senaryo

Haydi çocuklar! Bugün hayal gücümüzü kullanacağız. Hepiniz büyük bir robot fabrikasında stajyer mühendis oldunuz. Bu fabrikanın en yeni ve en akıllı robotu biraz önce üretim bandından çıktı. Adı: Orbit! 🤖

Göreviniz: Orbit'in ilk kimlik kartını hazırlamak. Bunu yapmak için Python dilinde **değişkenleri** ve **veri türlerini** kullanacağız.

Değişkenler ve Veri Türleri



Görev

Öncelikle Orbit hakkında bildiklerimizi kaydetmemiz gerekiyor.

- **Adı:** Orbit
- **Modeli:** XR-25
- **Yaşı:** 3 yıl
- **Boy:** 1.78 metre

```
# 🤖 Görev: Yeni gelen robotun temel bilgilerini  
kaydedin.
```

```
robot_adi = "Orbit"  
robot_model = "XR-25" # yıl  
robot_yasi = 3  
robot_boyu = 1.78      # metre  
  
print("🤖 Robot Tanıtımı")  
print("=====")  
print("Adı:", robot_adi)  
print("Modeli:", robot_model)  
print("Yaşı:", robot_yasi, "yıl")  
print("Boy:", robot_boyu, "metre")
```



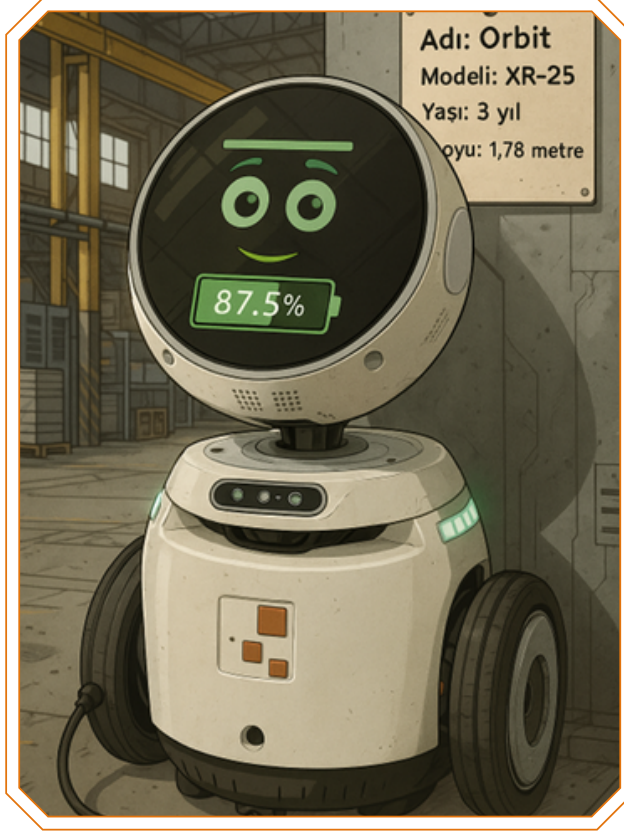
Kazanım

✅ str, int, float veri türleri tanınır.

Değişkenler ve Veri Türleri

Sahne 2

Pil Durumu ve Veri Tipi



Senaryo

Orbit'in kimlik kartını başarıyla hazırladık, harika iş çıkardın! 🎉

Ama bir sorun var: Orbit hareketsiz duruyor... 🤔 Neden mi? Çünkü daha **şarj durumu kontrol edilmedi!**

Robotlar da insanlar gibi enerjisiz kalınca çalışamaz. Onların enerjisi pillerinden gelir.

Bizim görevimiz, Orbit'in pil yüzdesini sisteme doğru bir şekilde kaydetmek. Ve bu bilgi **ondalıklı bir sayı** olacak.

🔍 Yani Python'da bu tür sayılara **float** denir. Noktalı ya da virgüllü sayılar!

Değişkenler ve Veri Türleri



Görev

- ✦ Orbit'in şarj seviyesi %87.5. Bu değeri **pil_yuzdesi** adında bir değişkene kaydet.
- ✦ Bu bilgiyi ekrana yazdır.
- ✦ `type()` komutunu kullanarak Python'dan bu sayının veri türünü öğren.

Hazırsan, kodunu yazmaya başlayabilirsin! 🚀

Bilmen Gerekenler:

💡 Float → Noktalı sayılar (örneğin: 87.5)

Python'da float (yüzen sayı) olarak adlandırılır

💡 `type()` fonksiyonu → Bir değer **veri tipini öğrenmek** için kullanılır

```
# 📁 Pil seviyesi bir sayı ama ondalıklı. 0
zaman float!
pil_yuzdesi = 87.5
print("📁 Pil Seviyesi:", pil_yuzdesi, "%")
# ✅ Veri tipi kontrolü
print("Pil yüzdesi tipi:", type(pil_yuzdesi))
```



Kazanım

- ✅ Noktalı sayıların Python'da float veri tipi olduğunu öğrendim 📦
- ✅ `type()` fonksiyonunu kullanarak bir verinin türünü nasıl bulacağımı keşfettim 🔍
- ✅ Değerin sadece ne olduğunu değil, nasıl bir veri olduğunu da sorgulamayı başardım 💡
- ✅ Orbit'in pil seviyesiyle veri tiplerini ilişkilendirerek konuyu daha iyi anladım ⚡🤖

Değişkenler ve Veri Türleri

Sahne 3

Orbit'in Görev Durumu – Aktif mi, Bozuk mu?



Senaryo

Orbit'in kimlik kartını oluşturduk ✓

Şarjını kontrol ettik ⚡

Şimdi sırada, onun **görev durumunu** ve **sağlık raporunu** sisteme kaydetmek var! 🤖

Bir robotun çalışıp çalışmadığını ve tamir gerekip gerekmediğini sadece “Evet” ya da “Hayır” şeklinde ifade ederiz.

Python'da bu tür bilgilere **mantıksal değer** denir. Yani ya **True** (doğru) ya da **False** (yanlış) olur.



Örnek Durum:

- Orbit şu anda görevdeyse → True
- Tamir gerekmiyorsa → False

Değişkenler ve Veri Türleri

```
# 🔍 Görevde mi? Tamire ihtiyaç var mı?  
gorevde_mi = True  
tamir_gerekliyor_mu = False  
  
print("🚗 Görevde mi? ->", gorevde_mi)  
print("🔧 Tamir gerekiyor mu? ->", tamir_gerekliyor_mu)
```



Kazanım

- ✅ Robotumuz çalışıyor mu? 😎 Yoksa tamire mi ihtiyacı var? 🔧 Bunlara evet/hayır şeklinde yanıt veririz. True ✅ ya da False ❌ değerleriyle!
- ✅ Bu tür bilgilere bool yani **mantıksal değer** denir! Sadece doğru veya yanlış! 😊

Değişkenler ve Veri Türleri

Sahne 4

Orbit'in Durum Raporu – Her Şey
Hazır mı?

Senaryo

Tebrikler mühendisler! 🧑‍🔧👩🧑‍🔧♂️

Orbit'in kimliği oluşturuldu, şarj seviyesi kontrol edildi ve görev durumu incelendi.

Şimdi sırada, bu bilgilerin hepsini bir araya getirerek **Orbit'in durum özetini** hazırlamak var.

Bu özet sistem ekranında temiz ve düzenli şekilde göstermeliyiz.

Python'da bunu yapmak için f-string yani **formatlı yazdırma** kullanabiliriz. Böylece değişkenleri yazıların içine kolayca yerleştiririz.

💡 Örnek Durum:

- Orbit şu anda görevdeyse → True
- Tamir gerekmiyorsa → False

Değişkenler ve Veri Türleri

```
robot_adi = "Orbit"  
robot_model = "XR-25"  
robot_yasi = 3 # yıl  
robot_boyu = 1.78 # metre  
pil_yuzdesi = 87.5  
gorevde_mi = True  
tamir_gerekliyor_mu = False
```

```
print(f"")  
Robot Özeti  
-----  
🤖 Adı: {robot_adi}  
🔧 Modeli: {robot_model}  
🕒 Yaşı: {robot_yasi} yıl  
📏 Boyu: {robot_boyu} m  
🔋 Pil: {pil_yuzdesi}%  
🚚 Görevde mi?: {gorevde_mi}  
🔧 Tamir Gerekliyor mu?: {tamir_gerekliyor_mu}
```



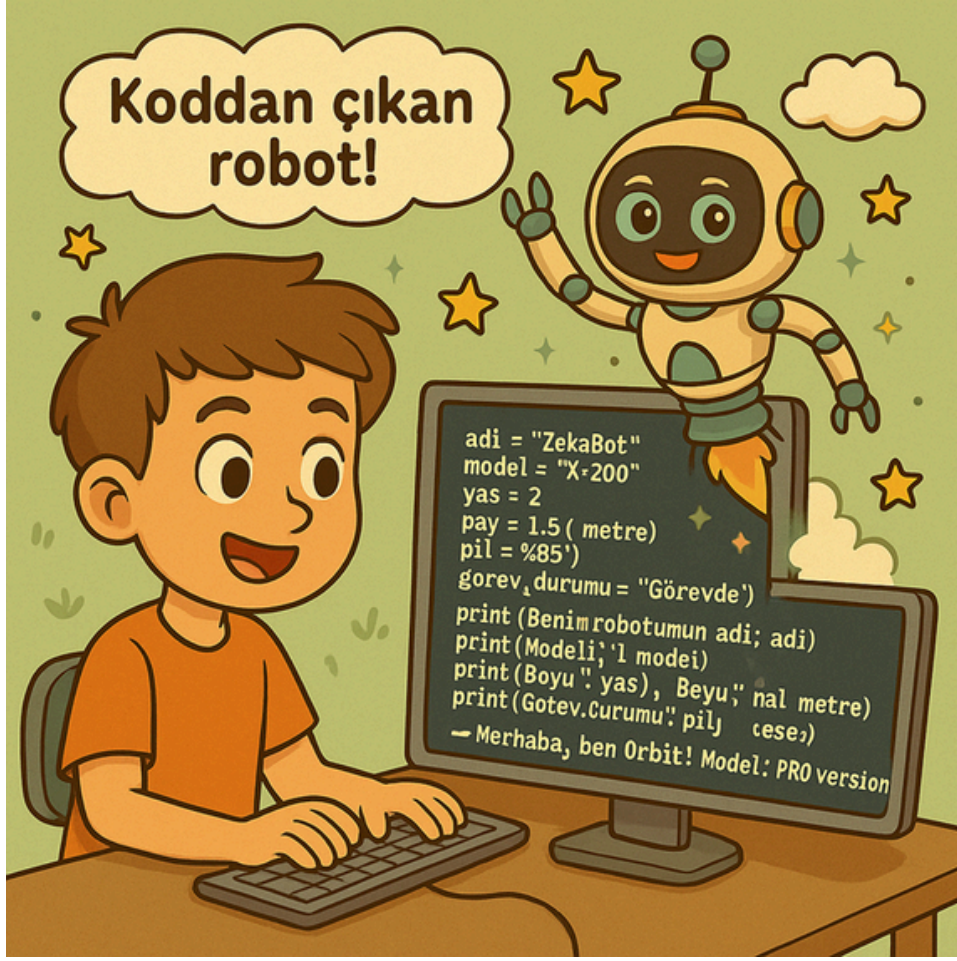
Kazanım

✅ Vay canına! Tüm bilgiler tek ekranda! f-string sayesinde hem düzenli hem de havalı görünüyor.

✅ Python'a "Bu yazıyı yaz ama aralara kutulardaki bilgileri yerleştir" diyoruz.

Yani `abc` + 📦 = 📄 profesyonel özet!

Değişkenler ve Veri Türleri



Öğrenciye Soru

🤖 "Senin hayalindeki 🤖 nasıl olurdu? Adı, modeli, yaşı, boyu, pili, görev durumu ne olurdu?"

Bunları değişken olarak tanımlayıp ekran çıktısı hazırla!"



Hedefler

- 📦 Değişken nedir, neden kullanılır?
- 📄 str, 📊 int, 📏 float, ✅❌ bool
- 🔍 type() fonksiyonu
- 🌟 f-string ile biçimli yazdırma